

University of New Orleans
ScholarWorks@UNO

University of New Orleans Syllabi

Fall 2015

CSCI 4670

Stephen G. Ware
University of New Orleans

Follow this and additional works at: <https://scholarworks.uno.edu/syllabi>

This is an older syllabus and should not be used as a substitute for the syllabus for a current semester course.

Recommended Citation

Ware, Stephen G., "CSCI 4670" (2015). *University of New Orleans Syllabi*. Paper 182.
<https://scholarworks.uno.edu/syllabi/182>

This Syllabus is brought to you for free and open access by ScholarWorks@UNO. It has been accepted for inclusion in University of New Orleans Syllabi by an authorized administrator of ScholarWorks@UNO. For more information, please contact scholarworks@uno.edu.

CSCI 4670: Fundamentals of Game Development

Spring 2015: Tuesday and Thursday in Math 122 from 2:00 to 3:15 PM.

Contents

- [Course Description](#)
- [Learning Outcomes](#)
- [Instructor](#)
- [Prerequisites](#)
- [Text](#)
- [Grading](#)
- [Projects](#)
- [Game Development Tools](#)
- [Policies](#)
- [Academic Integrity](#)
- [Students with Disabilities](#)
- [Calendar](#)

Course Description

This is the first class specifically focused on video game development. It covers basic principles of game design, game history, the games industry, and technical topics specifically related to games. The projects in this class are designed to increase your awareness of and familiarity with game design tools and to help you establish a very modest portfolio.

Topics covered in this class include:

- The definitions of concepts like game, toy, sport, play, flow, and engagement
- History of video games
- Game genres
- Core game mechanics
- Narrative
- Game physics
- Simple Artificial Intelligence
- Procedural Content Generation
- The nature of the video games industry
- Current game development tools and game engines
- Game design documents and pitches

This class is divided into two halves. The first half will focus on theoretical questions of game design, and the second will focus on practical game development programming skills.

This class will be most effective if you later take CSCI 4675: Advanced Game Development.

Learning Outcomes

After taking this class, students should:

- Understand game design and the history of video games in particular.
- Be able to write critically about games in terms of their rules, play, and culture aspects.
- Be familiar with basic game programming concepts, such as physics, AI, and content generation.
- Be familiar with at least three tools used to design and create games.
- Have a basic understanding of the nature of the game development industry.

Instructor

Prof. Stephen G. Ware
Department of Computer Science
Math Building, Room 337
sgware@uno.edu
Office Hours: 3:30 PM to 5:00 PM on Tuesday and Thursday

Prerequisites

Before taking this class, you must:

- Learn basic programming skills.
- Familiarize yourself with the Java programming language and the [Eclipse IDE](#).
- Pass *CSCI 2125: Data Structures* with a C or better.

Text

Rules of Play: Game Design Fundamentals (RoP for short) by Katie Salen and Eric Zimmerman

The book will primarily be used in the first half of the class and will be a major source of information for the written midterm exam. Reading assignments will come many chapters at a time, and I understand that you may not always have time to do the reading. Please *at least* read the 1 page summaries at the end of the chapters. To encourage the reading, I will give bonus points when someone answers an in-class question that demonstrates a clear understanding of the assigned reading.



Grading

Final grades will be determined by the following elements. The weight of each element depends on whether you are enrolled in the undergraduate or graduate version of the class.

Projects

Game Development Projects

If you plan to apply for a job in the games industry, you will need a portfolio. To that end, this class requires 3 small game design projects which are meant to demonstrate what you have learned, build your portfolio, and introduce you to current game development tools.

Choose 3 of the 5 projects listed below. You may do them in any order, but they are listed in a recommended order. You should use one or more of the development tools listed for that project, and you may not use the same development tools for more than one project. For example, if you use Unity to create a Puzzle Game, you may not also use Unity to create a First Person 3D Game. Other development tools not listed below must be approved by the instructor at least a week in advance of the project due date.

- **Physical Board or Card Game** using paper, card stock, markers, game pawns, 3D printing, etc.
- **Adventure Game** using Inform, Twine, or Adventure Game Studio
- **Puzzle Game** using Game Maker Studio, Torque 2D, Unity, HTML5, Java, or Objective C
- **Arcade Game** using Game Maker Studio, Torque 2D, Unity, HTML5, PyGame
- **3D Action Game** using Unity, Unreal Engine, Torque 3D (Up to 5% bonus can be earned for completing this project.)

These projects are *not* intended to be large undertakings, but simply to introduce you to new technology. Rather than creating a

game entirely from scratch, I highly recommend that you take an existing game or game tutorial and modify it in some way. Some examples of good projects would be:

- A simple “escape the room” adventure in Inform.
- Single player Tic-Tac-Toe Android app that is played on a 4x4 grid rather than a 3x3 grid.
- Complete the Unity stealth game tutorial, but modify the layout of the level.

Game Play Journals

One key skill taught in this class is the ability to critically evaluate a game. You will complete 3 Game Play Journals, during which you play a game (new or one you have played before) and write about your experience. These should be short--between 2 and 5 pages. These journals are meant to exercise your ability to analyse game design and provide constructive criticism to other designers.

Game Pitch Project

In place of the final exam, the last assignment of the semester will be to pitch a game concept to the instructor and your classmates. You do not need to actually make this game, only imagine it. You will create a pitch document that follows a specific template and give a short presentation in class to convince a fictional publisher to fund your idea.

Game Development Tools

[Inform 7](#) is a tool for creating text-based interactive fiction games using a syntax that reads like natural language. It has an excellent [video tutorial](#) and many other well-documented tutorials and examples built into the tool.

[Twine](#) is a tool for creating choose-your-own adventure stories. It requires little to no knowledge of programming. Anna Anthropy has written a very straight-forward [tutorial](#) for making simple games.

[Adventure Game Studio](#) is a mature, full-featured engine for creating graphical point-and-click adventure games. Most modern graphic adventures (professional and amateur) are made with this tool. It has a comprehensive set of [video tutorials](#).

[HTML5](#) is the latest standard for the website markup language HTML. One of its most prominent features is the [canvas element](#), which allows for the creation of dynamic 2D graphics with Javascript. Games made in HTML can be played on many platforms---virtually anything with a web browser. [This tutorial](#) is a good place to start for making games. [This tutorial](#) is a little more comprehensive.

[Game Maker Studio](#) is a general, easy-to-use tool designed specifically for making 2D games. It has a graphical point-and-click interface for making game logic, which means you can create games without using any code, however it also has a simple and very functional scripting language. It comes with [tutorials](#) on how to use specific features, but also a number of [walkthroughs](#) for creating sample games.

[Torque 2D](#) and [Torque 3D](#) are a pair of open source, full-featured game engines. There are [a handful of tutorials for Torque 2D](#) that walk you through making a sample game. There are also a detailed set of [tutorial videos for making a multi-player first person shooter in Torque 3D](#).

[Unity](#) is a popular 3D (and now also 2D) game engine. It is favored by independent developers, but is quickly gaining traction at major studios. It has a number of [tutorials](#) for individual skills and whole example projects.

The [Unreal Engine](#) has long been the industry standard for the development of 3D games by major studios. It has a high learning curve, but is extremely powerful. Unreal 4 has provided many significant improvements to the documentation and usability of the engine, including [many helpful tutorials](#).

Policies

These policies are in place to maintain professionalism and mutual respect:

- **Attendance:** Attendance is not required, but it is always strongly encouraged.
- **Devices in Class:** No electronic devices, such as laptops, tablets, or phones, may be used during class unless there is an in-

class programming assignment scheduled for that day. You do not have to come to class, but if you do, you have to be a polite and attentive audience.

- **Phones:** Please silences your phones before class.
- **Due Dates:** All assignments are due on Moodle promptly at midnight on the day they are due; however, the submissions will not close until 3 AM the next morning. Consider these three hours an unofficial grace period. At 3:01 AM, you assignment is not 1 minute late; it is 3 hours and 1 minute late, and it will not be accepted.
- **Late Work:** Late assignments *will not be accepted* without a serious and documented excuse.
- **Missed Exams:** If you need to miss an exam, you must notify me at least a week in advance to schedule a makeup. Missed exams cannot be retaken without a serious, documented excuse.
- **Food and Drink:** Please do not bring food to class. Drinks are allowed, as long as you are respectful of your classmates.
- **E-mail Response Time:** Please allow a minimum of 24 hours for e-mail responses.

Academic Integrity

All students are expected to follow [UNO's code of academic integrity](#). Violations of this policy will be dealt with on a case-by-case basis, but punishment will be severe and may include failing the class and expulsion from the university.

Students with Disabilities

It is university policy to provide, on a flexible and individualized basis, reasonable accommodations to students who have disabilities that may affect their ability to participate in course activities or to meet course requirements. Students with physical or learning disabilities should contact the instructor and the [Office of Disability Services](#). Arrangements will be made on an individual basis.

Calendar

Topic

Thursday, August 20
Introduction and Syllabus

Game Design

Tuesday, August 25
Meaningful Play

Thursday, August 27
Defining Games

Tuesday, September 1
History of Video Games

Thursday, September 3
History of Video Games (cont.)

Sunday, September 6
Project Deadline

Tuesday, September 8
History of Video Games (cont.)

Thursday, September 10
Rules

Tuesday, September 15
Rules Schemas: Emergent Systems and Uncertainty

Thursday, September 17
Rules Schemas: Information Theory and Information Systems

Assignments

- Review the syllabus.
- Read chapters 1 to 6 in RoP.
- Read chapters 7 to 10 in RoP.
- First Game Play Journal due on Moodle at midnight.
- First Game Development proposal due on Moodle at midnight.
- Read chapters 11 to 13 in RoP.
- Read chapters 14 to 15 in RoP.
- Read chapters 16 to 17 in RoP.

Tuesday, September 22

Rules Schemas: Game Theory, Cybernetic Systems, Conflict

Thursday, September 24

Play and Play Schemas: Experience, Meaning, Pleasure

Sunday, September 27

Project Deadline

Monday, September 28

Project Deadline

Tuesday, September 29

Play Schemas: Narrative, Simulation, Social

Thursday, October 1

Culture Schemas: Rhetoric, Open

Sunday, October 4

Project Deadline

Tuesday, October 6

Culture Schemas: Resistance, Environment

Thursday, October 8

Rules of Play Wrap-Up

Tuesday, October 13

Midterm Exam

Thursday, October 15

Mid-Semester Break

Tuesday, October 20

Metagaming

Game Development

Thursday, October 22

Game Physics

Sunday, October 25

Project Deadline

Monday, October 26

Project Deadline

Tuesday, October 27

Game Physics (cont.)

Thursday, October 29

Game Physics (cont.)

Sunday, November 1

Project Deadline

Tuesday, November 3

Game AI: A* Pathfinding

Thursday, November 5

Game AI: Behavior Trees

Tuesday, November 10

Game AI: Behavior Trees (cont.)

Thursday, November 12

Game AI: MiniMax Search

Tuesday, November 17

Procedural Content Generation

- Read chapters 18 to 20 in RoP.

- Read chapters 22 to 25 in RoP.

- First Game Development deliverable due on Moodle at midnight.

- First Game Development post-mortem due on Moodle at midnight.

- Read chapters 26 to 28 in RoP.

- Read chapters 29 to 31 in RoP.

- Second Game Development proposal due on Moodle at midnight.

- Read chapters 32 to 33 in RoP.

- Second Game Play Journal assignment due on Moodle at midnight.

- Second Game Development deliverable due on Moodle at midnight.

- Second Game Development post-mortem due on Moodle at midnight.

- Third Game Development proposal due on Moodle at midnight.

- Third Game Play Journal assignment due on Moodle at midnight.

Thursday, November 19

The Games Industry

Sunday, November 22

Project Deadline

- Third Game Development project due on Moodle at midnight.

Tuesday, November 24

The Games Industry (cont.)

Thursday, November 26

Thanksgiving Holiday

Tuesday, December 1

Game Pitch Presentation Preparation

Thursday, December 3

Game Pitch Presentation Preparation

Thursday, December 10

Game Pitch Presentations
